CLAIMS

What is claimed is:

1. A method for resolving a partial media topology, comprising:

receiving a partial media topology that includes a plurality of nodes including at least one media source node and at least one media sink node;

populating a working FIFO queue with source nodes in the partial topology;

iteratively, for each node in the working FIFO queue:

negotiating a media type for each output of the node with the downstream node in the partial topology;

instantiating one or more intermediate nodes;

connecting the one or more intermediate nodes between the media source node and the media sink node; and

adding the one or more intermediate nodes to the working FIFO queue.

2. The method of claim 1, wherein the partial media topology is received from a remote process as a parameter in an interface call.

3. The method of claim 1, wherein the working FIFO queue comprises each node in the partial topology.

4. The method of claim 1, wherein negotiating a media type comprises determining the media types of an upstream node and an associated downstream node.

5. The method of claim 1, wherein instantiating one or more intermediate nodes comprises instantiating at least one of an encoder or a decoder.

6. The method of claim 5, wherein the decoder converts a compressed output stream of the source node into an uncompressed output.

7. The method of claim 5, wherein the encoder converts an uncompressed media stream into a compressed media stream.

8. The method of claim 1, wherein connecting the one or more intermediate nodes between the upstream node and the downstream node comprises generating a data path between the output of a upstream node an input of an intermediate node.

9. The method of claim 1, wherein one or more of the intermediate nodes is an option node.

10. A system comprising:

one or more computer-readable media;

a media engine embodied on the one or more computer-readable media and configured to communicatively interact with an application to present a media presentation;

the media engine being configured to use:

a media session to generate a partial topology, the partial topology including one or more media sources individual ones of which serving as a source of media content, and one or more media sinks configured to sink a media stream, and

a topology loader to resolve the partial topology into a full topology.

11. The system of claim 10, wherein the media engine exposes one or more application program interfaces that are used by an application to interact directly with the media engine, and indirectly with components used by the media engine.

12. The system of claim 10, wherein the media session invokes the topology loader using an application programming interface.

13. The system of claim 10, wherein the media session passes the partial topology to the topology loader as a parameter in an interface call.

14. The system of claim 10, wherein the topology loader is configured to instantiate one or more intermediate nodes, and to connect the one or more intermediate nodes in a communication path between a media source and a media sink in a partial topology.

15. The system of claim 14, wherein the one or more intermediate nodes comprise a decoder for decoding the output of a source node.

16. The system of claim 14, wherein the one or more intermediate nodes comprises an encoder for encoding an input of a source node.

17. The system of claim 14, wherein the one or more intermediate nodes comprise an optional node, and wherein the topology loader implements logic to connect an optional node.

18. The system of claim 10, wherein the topology loader provides at least one interface to provide the application the capability to facilitate resolving the partial topology.

19. The system of claim 10, wherein the topology loader returns a fully resolved topology.

20. A system comprising:

one or more computer-readable media;

a media engine embodied on the one or more computer-readable media and configured to communicatively interact with an application to present a presentation;

the media engine being configured to use:

a media session to generate one or more media sources individual ones of which serving as a source of media content, and one or more media sinks configured to sink a media stream;

a topology loader to generate one or more transforms communicatively linked with one or more media sources and configured to operate on data received from the one or more media sources.

21. The system of claim 20, wherein

the media session is configured to first create partial topology that is to be utilized to present the presentation; and

the topology loader is configured to receive a partially resolved topology from the media session, and to generate a fully resolved topology.

22. The system of claim 21, wherein the media engine creates partial topology by at least determining one or more media sources and one or more media sinks for the presentation.

23. The system of claim 20, wherein the topology loader analyzes the outputs of a media source and the inputs of a media sink, and negotiates the media type for passing a media stream between the media source and the media sink.

24. The system of claim 20, wherein the topology loader generates a source node list comprising nodes in the partial topology.

25. The system of claim 24, wherein the one or more transforms generated by the topology loader are added to the source node list.

26. The system of claim 25, wherein the topology loader negotiates the media type between the one or more transforms and one or more downstream nodes.

27. The system of claim 20, wherein the one or more transforms comprises at least of an encoder or a decoder.

28. The system of claim 20, wherein the topology loader returns the fully resolved topology to the media session.